# Jalapeno: A New, High-Performance x86 Core

**Greg Grohoski**

**Cyrix Corporation**
**A National Semiconductor Company**

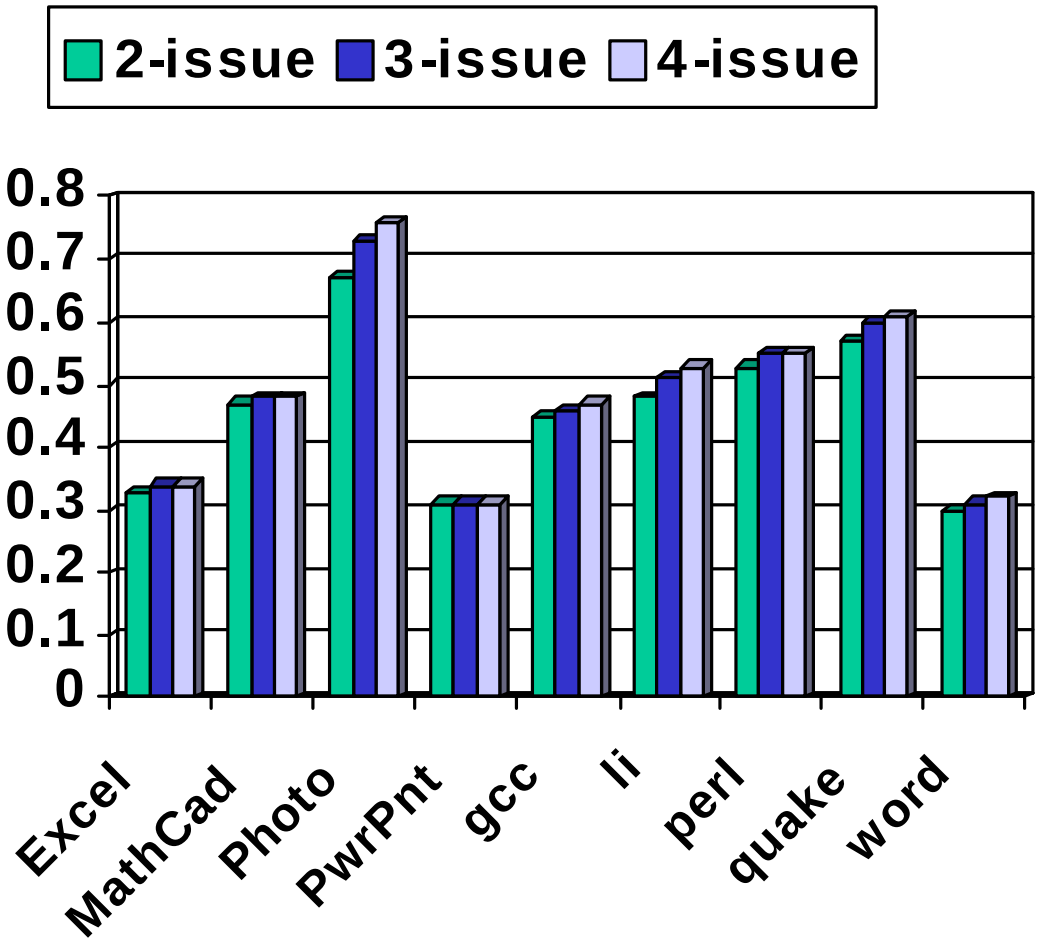

# Design Objectives

- High frequency
- Reduce memory latency
- Improve floating-point and 3D performance
- Small die size

# Observations

- **Typical PC applications spend ~40-60% of time in OS**
- **OS has limited parallelism**
  - Data-dependent conditional branches, pointer-chasing loops
  - Frequent context switches (cache, TLB, BTB)
- **Some parallelism in games (3DNow)**
- **Amdahl's law - Memory latency limits performance potential of wide-issue designs**

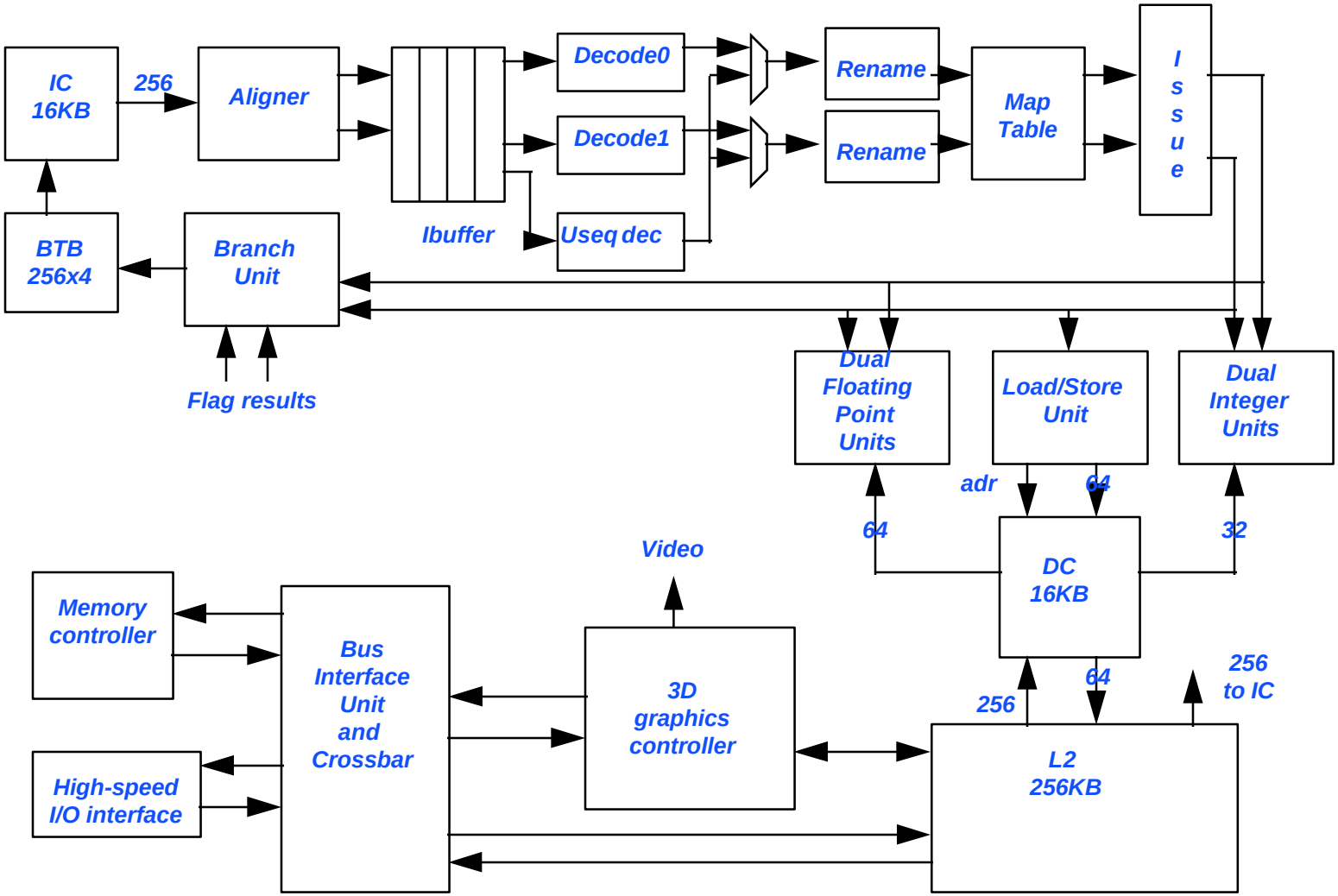
# Issue width vs. ipc



# Design philosophy

- **Dual-issue is best area/performance trade-off**
  - Exploit parallelism of newer applications
  - Minimize area cost, complexity, and frequency limitations of wider-issue machines
- **Utilize area to:**
  - Minimize memory latency
  - Integrate PC components:
    - Improve overall system performance
    - Reduce cost, system footprint, and power dissipation
- **Deliver outstanding performance and value**

# M3 block diagram



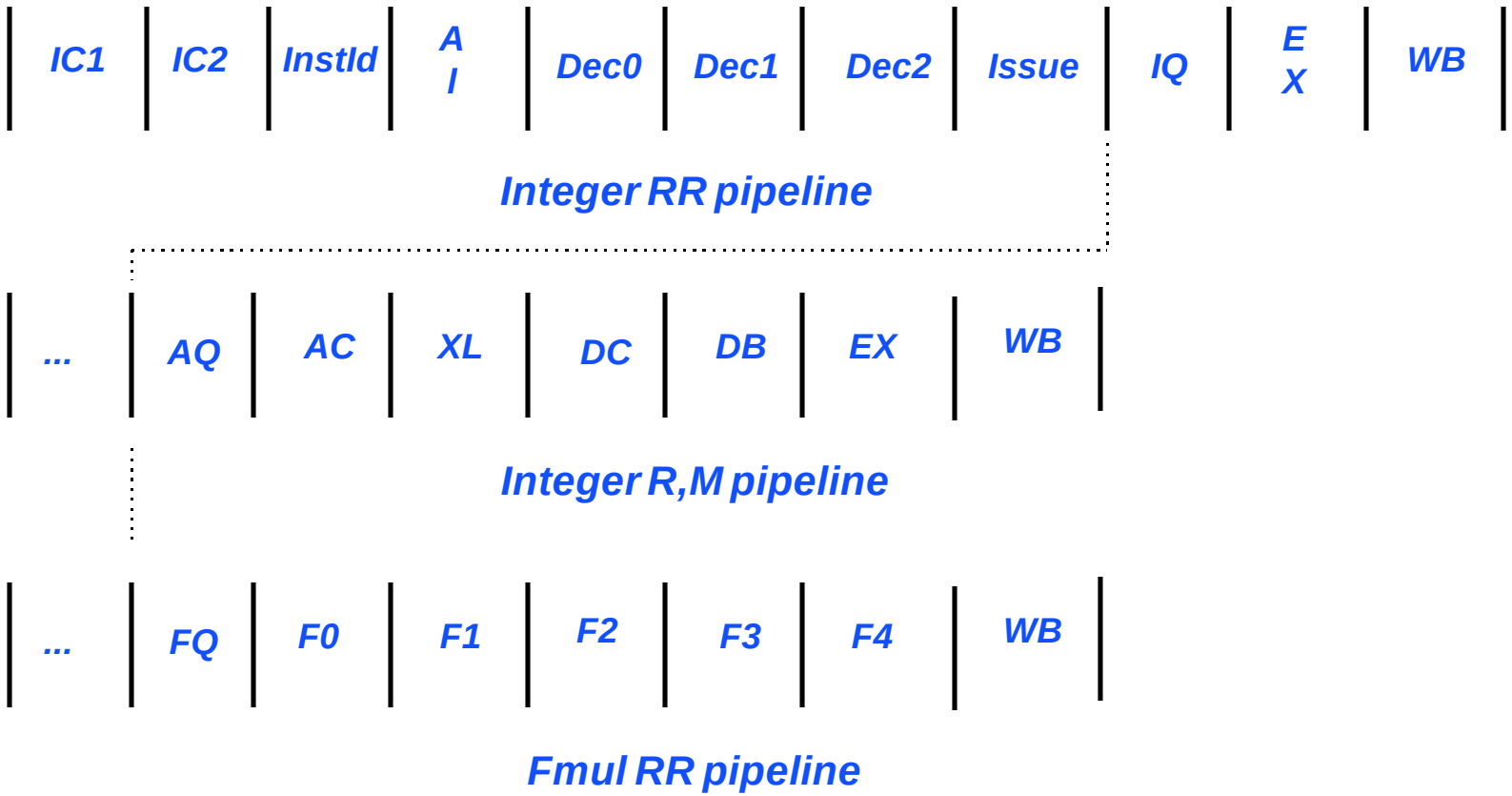

National Semiconductor ®

Cyrix® A National Semiconductor Company

# Jalapeno x86 core

- **Dual-issue x86 with register renaming and out-of-order execution**
  - 2 integer units
  - 2 FP/MMX units
  - 1 branch unit
  - 1 load/store unit
- **Deep pipeline - 11 stages fetch to integer completion**
  - 600+ MHz in 0.18u
- **Branch prediction**
  - 1K entry, 4-way BTB with 7-bit history and prediction ROM
  - 16 entry return stack
- **16KB, 4-way IC, 2 pending misses**
- **16KB, 4-way, dual-ported DC, 4 pending misses**
- **256KB 8-way L2, 8 pending misses**

# Instruction Pipelines



# Instruction Decoding

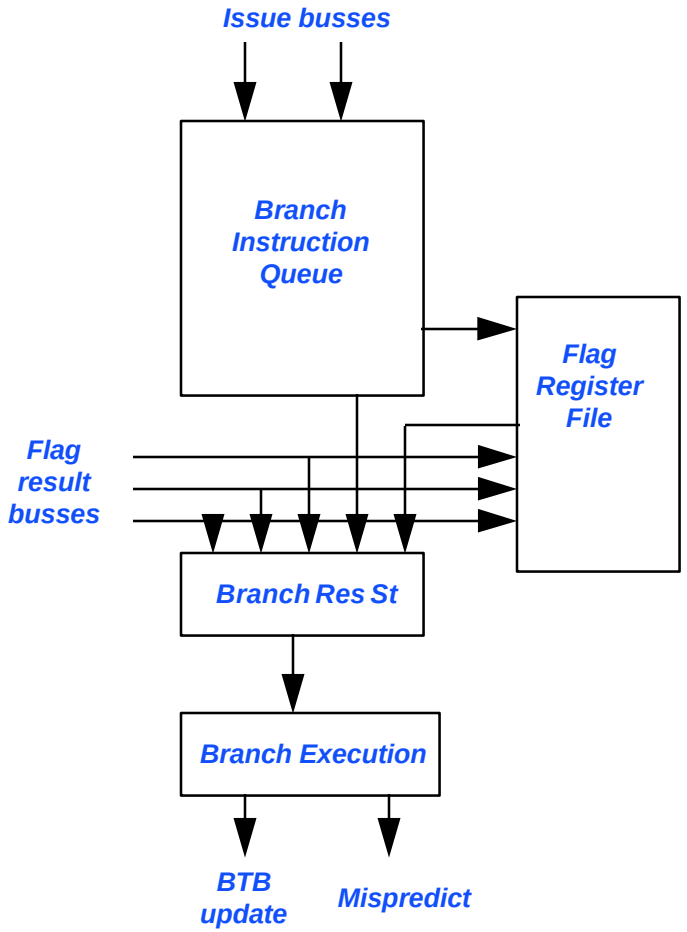
- **Next fetch address is predicted from BTB**
- **32 bytes are fetched from the IC and buffered**
- **Up to 2 full x86 instructions are identified & buffered (up to 2 prefix bytes)**
  - No cached predecode bits
- **Decode & issue constraints are applied**
  - Only 1 memory op/cycle
- **Instructions are decoded into ‘nodes’**

# Node Issue

- **Instructions map to 1, 2, or more nodes**
  - Each node contains source, destination, and control
- **Up to 2 EX, 1 AC, 1 BR, and 2 FPU nodes can be issued in parallel**
- **Most instructions can be dual-issued**
  - padd reg, mem / pand reg, reg
  - fadd reg, reg / fmul reg, reg
  - add mem, reg / shl reg, reg
  - cmp reg, mem / jcc
- **Nodes are formed into checkpoints, renamed, and issued to execution units**
- **Up to 16 checkpoints active (96 nodes)**

# Branch processing

- Separate branch unit - more core bandwidth
- Flag results generated out-of-order
- Executes branches in-order
- Resolves mispredicts at branch execute time and begins recovery fetch
- Typical misprediction recovery is 12 clocks

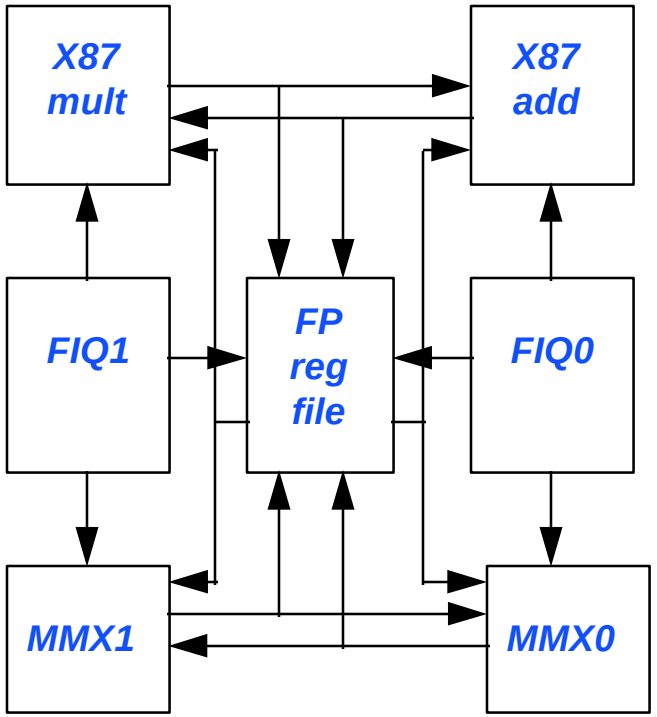


# Integer Units

- **Fully pipelined**
- **Nearly all ops execute and bypass results in 1 cycle to both units**
  - RCL / RCR / BSF / BSR are 2 cycles
  - Integer multiplies are 8-12 cycles (execute in MMX multiplier)
  - Integer divides - 1 quotient bit per cycle with early out
- **Parallel execution of integer ops with integer multiplies and divides**
- **Fast integer <-> FPU/MMX result forwarding**

# FP / MMX unit

- **Fully pipelined x87 adder, 4/1**
- **Fully pipelined x87 multiplier, 5/1**
- **FDIV 23 (SP), 33 (DP, EP)**
- **Dual execution units support MMX and 3DNow**
  - MMX: 1/1; shift 2/1; mul 4/1
  - 3DNow: 3/1; pfrcp/pfsqrt/pfmul 5/1



# Load / Store unit

- **Single load/store unit for efficiency**
  - 16KB, 4-way, non-blocking data cache (3 cycle access, 1 load port, 1 store/fill port)
  - 32-entry, fully-associative L1 DTLB
  - 512-entry, 8-way L2 TLB
  - 12 entry store queue with data forwarding
- **In-order translation of requests**
- **Up to 4 pending L1 misses**
  - Miss data returns out-of-order
  - Hit-under-miss
- **Data is supplied in-order to execution units**

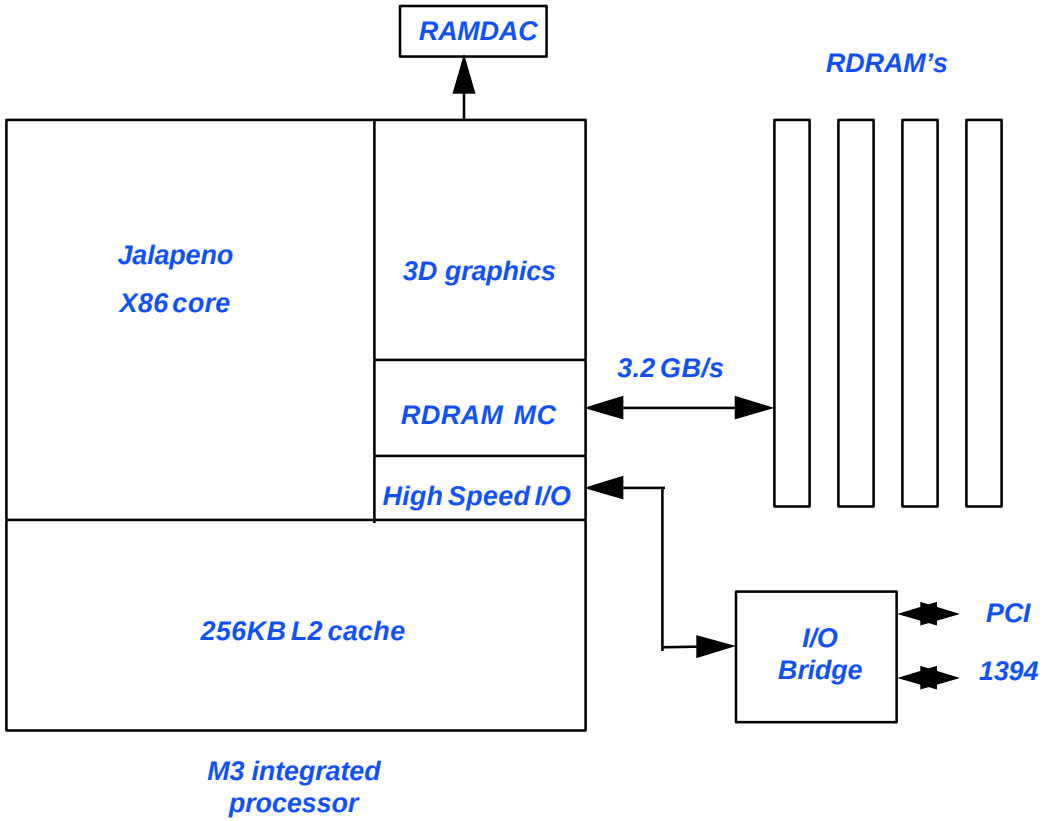
# L2 cache

- **256KB, 8-way associative, 8-way interleaved**
- **Each interleave supports**
  - 1 L1 miss per cycle
  - 1 L1 store / L2 fill per cycle
- **Fully pipelined at core frequency**
- **7 clock access latency from L1**
- **256 bit refill to L1 (no trailing-edge)**
- **Sets can be locked down for virtualization code or use by graphics engine**

# Reducing Memory Latency

- **I-cache prefetches from L2, 2 pending misses**
- **D-cache prefetches from L2, 4 pending misses**
- **L2 prefetches I-misses and D-misses independently from DRAM**
  - 25 to 50% of L2 misses are predictable
  - Prefetching minimizes delays for compulsory misses
  - 256KB cache performs like 512KB
- **Core -> DRAM access time is minimized via on-chip memory controller**
  - <20 nS overhead to critical word vs. 50+ nS via conventional socket-7/slot-1 bus (excluding DRAM access time)
  - Memory controller keeps 32 open pages - much greater than the average chipset

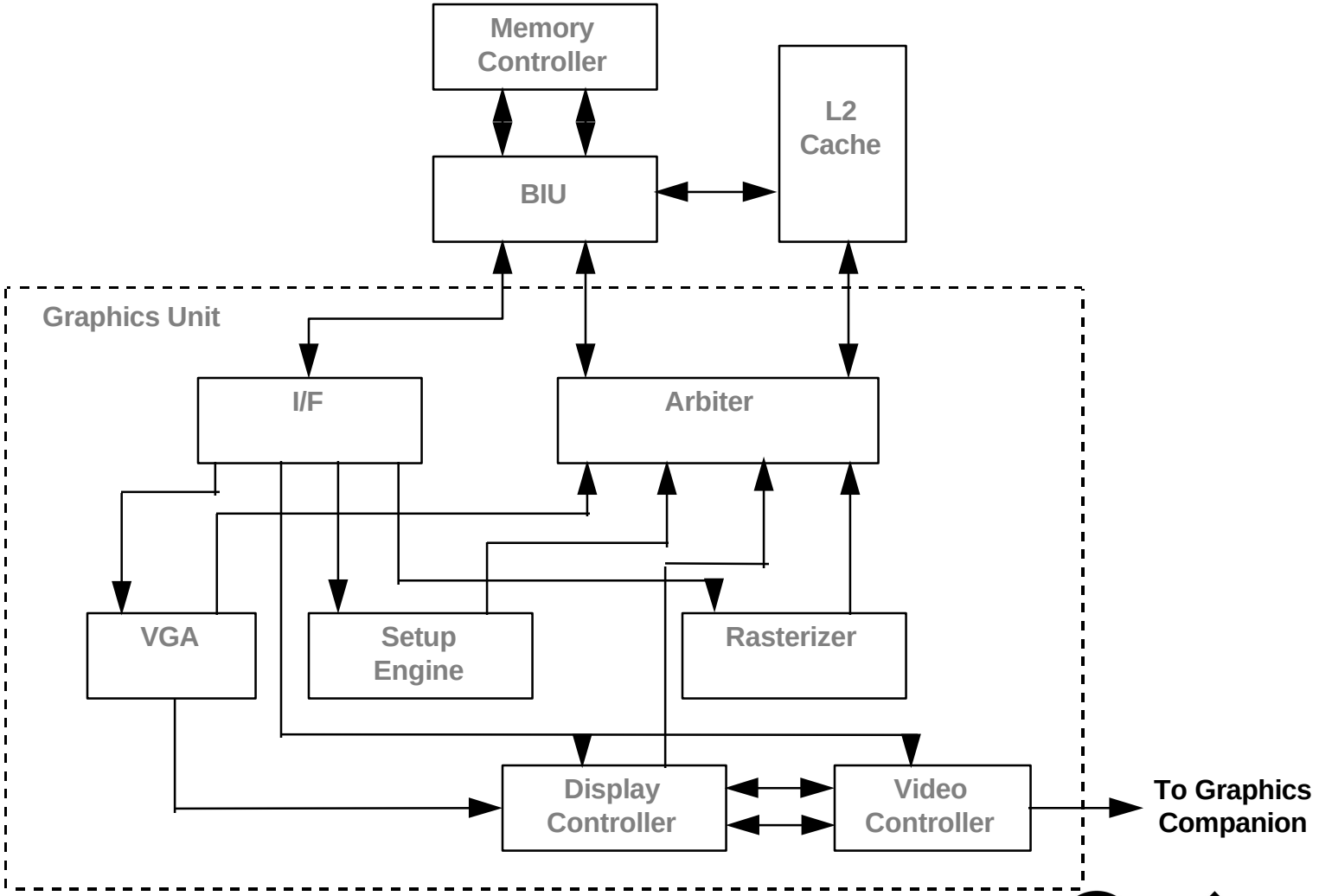
# Example M3 Integrated Processor System Diagram




National Semiconductor ®

Cyrix® A National Semiconductor Company

# Graphics Subsystem



# Advantages of Integrating Graphics

- **Use L2 cache for graphics functions**
  - Texture caching
  - Composite buffer for multipass graphics features
- **Unified Memory Architecture**
  - Reduces system cost by eliminating dedicated Frame Buffer memory
  - Allows fetching textures out of GART space with no additional latency vs. Frame Buffer fetches
- **Tightly coupled memory controller interface**
- **Leverage CPU design techniques and circuits**

# Graphics Performance and Features

- **Performance**
  - 3M Polygons/sec
  - 266M Pixels/sec
  - 230MHz dot clock
- **Features**
  - Industry standard API support
  - Independent Fog/Alpha
  - Anisotropic Texture Filtering
  - Anti-aliasing
  - Integrated MPEG2/DVD Playback

# Summary

- **Jalapeno**
  - Compact, low-cost, 2-issue x86 core
  - Memory-centric
  - Deep pipeline for high-frequency operation - 600+ MHz
- **M3**
  - Advanced 3D graphics
  - High performance, reduced latency memory subsystem
  - 120 mm**2 in 0.18u technology
  - Samples 4Q99